Pentest Report

Table of Contents

Removed / Redacted Information: Emails, Company Links, Screenshots, & the Web Devolper's Github Account.

1 Preface

This report will reference tests which provided tangible results, but not an exhaustive list of all actions performed. The report will be structured as follows:

- **Scope**: Outline of the assets tested, and which tests were omitted.
- **Executive Summary**: A high-level overview of the findings, followed by a detailed description of the vulnerabilities identified and other potential risks posed to the company.

- **Conclusions**: Recommendations to improve security, ways in which testing could have been improved, and additional tests which could be performed at a later date.
- **Appendicies**: Links to resources for further reading (i.e. CVEs and background information on the vulnerabilities referenced).

2 Scope

Tests were performed on the domains **DOMAIN.com** and **DOMAIN2.com**. No limitations were provided by the client. To minimize the risk of disruptions, Denial of Service (DoS) Attacks and Social Engineering (e.g. Phishing) were not evaluated; these attack vectors can be examined at a later date based upon the client’s discression. I cannot perform tests pertaining to integrated payment processors (i.e. Paypal), active tests involving resources which belong to third parties (e.g. contracted web developers), or accounts on third party applications (e.g. Gmail).

2.1 Engagement

A Blackbox Penetration Test was performed; ergo, no information pertaining to the target’s infrastructure was provided and no credentials for testing accounts were given. The only provided information was the email address **CLIENT@DOMAIN.com**.

Tests were performed from an external IP address, beginning on *07/29/2024* and ending on *08/20/2024*.

3 Executive Summary

3.1 Overview

I found two high severity vulnerabilities and three less severe vulnerabilities which could be leveraged in conjunction with user interaction and/or by an attacker who is able to sign into the website. I also identified credentials associated with company emails that were exposed in data breaches.

The vulnerabilities I identified were the result of certain WordPress Plugins. A Plugin is a third-party add-on that can be installed to modify the functionality of a WordPress website. For example, a Plugin can be used for cosmetic changes or to add extra features, like a CAPTCHA for forms that handle user input.

Out of the five vulnerable Plugins identified, there were three types of vulnerabilities present: XSS, CSRF, and Information Disclosure. These vulnerabilities can be chained together for a greater impact.

For example, an unauthenticated attacker may be able to gain access to an Administrator account by leveraging either the CSRF or the Reflected-XSS vulnerability. The hacker would then have full control over the site, and could view the password for an email account via the Information Disclosure vulnerability.

3.2 Vulnerabilities

3.2.1 Cross-Site Scripting (XSS)

XSS occurs when an attacker is able to insert malicious code in a website's URL (e.g. Reflected-XSS) or embed it within one of the site's Web Pages (e.g. Stored-XSS). The code will be automatically executed by a user's Web Browser when they click the malicous URL or visit the compromised Web Page; the effect of a XSS payload will rarely be visible to the affected user.

One of the worst case scenarios would involve an attacker leveraging XSS to steal all insecure cookies in a user's browser session, including those from other websites. A stolen cookie allows an attacker to sign into the corresponding website, without needing to know the account's username or password.

Other potential XSS attacks include, but are not limited to, redirecting the user to another site, or modifying the presentation of a website's content.

3.2.1.1 white-label-cms Plugin

The version of the **white-label-cms** installed is vulnerable to Reflected-XSS. This vulnerability is considered high severity due to the ease of exploitation, and since it can be exploited by an unauthenticated attacker.

3.2.1.2 Slider Revolution Plugin

The version of the **revslider** (aka Slider Revolution) Plugin installed is out-of-date, making it vulnerable to Stored-XSS. However, in this case an attacker could not exploit it unless they were already able to sign into an account with Administrator access. This prerequisite significantly decreases the severity of this vulnerabilty.

3.2.1.3 filr-protection Plugin

The version of the **filr-protection** Plugin installed is vulnerable to Stored-XSS. This is only exploitable for authenticated attackers with at least editor-level permissions, which decreases the severity of this vulnerability.

3.2.2 Cross-Site Request Forgery (CSRF)

Whenever users perform an action on a website, such as clicking a button, this is translated into code behind the scenes. In the case of a CSRF attack, an attacker is able to execute code which mimics user input in the context of the victim’s browser session. Put simply, any action that a user is able to perform on a website can be automatically, and imperceptibly, performed. Similarly to XSS, a CSRF attack requires some form of user interaction to trigger, such as visiting a web page with embeded malware.

The impact of a CSRF attack varies based upon the privileges of the victim’s account. For instance, if an Administrator is the target of a CSRF attack, the attacker may be able to completely compromise the website by creating an additional Administrator account that uses credentials which they specified. In the case of a WordPress site such as this one, an attacker could also add or modify a Theme or Plugin to get Remote Code Execution (RCE) on the underlying Web Server. Both of these examples give a hacker total control over the entire website.

3.2.2.1 contact-form-7-style Plugin

The **contact-form-7-style** Plugin is deprecated, meaning it is no longer available for download and updates are no longer being issued. Your site uses version **3.1.9** which contains a high severity CSRF vulnerability. There was an attempted patch in version **3.2.0**, but this proved ineffective. This means that there are no secure versions of this software, and it will not be fixed.

An unauthenticated attacker can embed the CSRF payload inside of a form field on your website's contact page. This payload would then be triggered by the user who viewed this message.

https://DOMAIN.com/contact-us/

3.2.3 Information Disclosure

An Information Disclosure vulnerability means that some form of sensitive information, such as passwords, can be viewed by people who should not have access to it.

3.2.3.1 wp-mail-smtp Plugin

The version of the **wp-mail-smtp** Plugin installed is out-of-date, making it vulnerable to Information Disclosure. In this case, an attacker who is signed into an Administrator account can view the configured SMTP password. Put simply, an attacker with the prerequisite access will be able to read the password for the email account that your website is setup to use.

This vulnerability is considered less severe, since it requires a high level of access to leverage it. However, should this be exploited, there is a risk that a hacker could access resources beyond the scope of the website itself; additional information on the potential impact of compromised email accounts will be provided in a subsequent section.

3.3 Compromised Credentials

This section of the report has nothing to do with a vulnerability in the website itself, however it still describes a potential risk to your company. There is a dangerous tendency for people to reuse passwords across multiple accounts on unrelated platforms that they use in a buisness or personal setting.

Even if someone does not reuse an identical password, they will often use a very similar one; it is common practice for hackers to generate common permutations of idenfitied passwords in an attempt to brute force logins. Oftentimes these mutations include changing the capitalization of letters, adding numbers or special characters, etc.

Attackers will often test a list of passwords against any services connected to an individual, attempting to access: the target site, the victim's personal and work email addresses, any personal or buisness social media accounts, etc.

3.3.1 Obtaining Usernames

At the start of this engagement I was given the email address **CLIENT@DOMAIN.com**, through which I was able to infer that employee emails likely followed the syntax of **First_Name@DOMAIN.com**. I cross referenced this with a list of employees listed on the **/our-people** subdirectory to create a list of potential company email addresses. While scraping website content, I also found the email **DEV@WEBDEV.com**.

From there, I determined which of these emails correspond to user accounts on the **DOMAIN.com** website by leveraging WordPress' normal behavior. When you attempt to sign into WordPress website, you will get a different output whether you used a valid username with an invalid password, or an invalid username.

In addition to these, I found that the WordPress site has the **DOMAIN** user account, which can be found via a variety of methods; for example, this user is the author of every blog post on the website. In summary, I identified the following three WordPress users:

- **CLIENT@DOMAIN.com**
- **DEV@WEBDEV.com**
- **DOMAIN**

3.3.2 Data Breaches

Data Brokers aggregate the information obtained from Data Breaches, and then allow individuals to search through it for personal information (e.g. for passwords associated with a specified email address or username). This is how I was able to find passwords that were used at some point by various company email accounts.

This report will not include the credentials I obtained, but will list the accounts which I found credentials for and the source of the data leak; hopefully this will assist users in determining which passwords, if any, must be changed.

For more details on the data leaks which affected your account, see the following URL; this site will not disclose any leaked credentials.

https://haveibeenpwned.com

- **DEV@WEBDEV.com**: I found multiple unique passwords and a hash from numerous data breaches. Some of the aforemententioned credentials came from 123RF (2020), Houzz (2018), or were distributed on various hacking forums.

  The findings indicate a pattern of reusing similar passwords, which is especially troubling given the nature of this account, as described in the following section.

- **CLIENT@DOMAIN.com**: I found two passwords from two seperate data breaches. One from Zynga (2019) and another from Avvo (2019).

- **PARTNER@DOMAIN.com**: I found a password hash from Adobe (2013).

- **EMPLOYEE@DOMAIN.com**: I found a password from Avvo (2019).

3.3.3 Potential Impact

Access to any WordPress account could pose a security risk. As previously discussed, some vulnerabilities require authentication to exploit. This means that access to even a low privilege account could grant an attacker additional attack vectors. An attacker with access to an Administrator account will gain complete control over the website, and will be able to get Remote Code Execution (RCE) on the AWS Cloud Server itself.

Email accounts are often associated with other services. When logging into a website, such as LinkedIn, you may see a button that says something like “Continue with Google” or “Sign in with email.” This means of authentication is known as Single Sign-on (SSO). If an attacker has access to your email account, they can also sign into a corresponding account on a third party application which is setup to use SSO.

Even if an application is not setup to use SSO, it will still likely require that you provide an email address in case you lose your password; therefore, an attacker could simply request a password reset and leverage their access to your email in order to take over take over that account.

Furthermore, email is one of the primary mechanisms used by Two-Factor Authentication (2FA), which offers no protection against someone who has access to your email account.

Additionally, a compromised email account makes it easier for an attacker to Phish other employees. In a worst case scenario, Phishing can give a hacker access to the computer that the victim used to open the malicious email.

I researched the domain from the **DEV@WEBDEV.com** email address and determined that it belongs to the Web Developer presumably responsible working on the **DOMAIN.com** website. As such, it is likely that this account has elevated privileges on the site.

Furthermore, an attacker may also try to utilize the Web Developer’s compromised credentials to access their Github account, which I was able to locate based off of the profile’s name and the company logo. If there was a private repository used for the **DOMAIN.com** website, a hacker could: examine the Back End code, look for sensitive information in the commit history, and even attempt to poison the source code. These theories were not tested as they are out-of-scope. However, it underscores why it is important for this user to employ strong passwords on the **DOMAIN.com** website and on any tangentially related applications.

https://github.com/WEBDEV

4 Conclusions

4.1 Remediation

Any potential remediation steps provided should only be viewed as suggestions based upon a limited understanding of the client's buisness; security always involves trade-offs, so aspects such as convenience and the cosmetic design of a website should also be considered.

4.1.1 Patching Current Vulnerabilities

The severity of a vulnerability is estimated with a 0-10 score known as CVSS. This value is calculated based on factors such as: attack complexity, privileges required, potential impact, etc.

This is not a measure of the risk posed to your specific company, and in practice a vulnerability can still pose a significant risk even if it is assigned a lower CVSS score. However, a CVSS score may still help to prioritize vulnerability patches.

4.1.1.1 contact-form-7-style

There are no safe versions. Uninstall the Plugin.

CVSS: 8.8/10 (High)

4.1.1.2 white-label-cms

Version **2.7.4** is in use. Update to version **2.7.5** or later.

CVSS: 7.1/10 (High)

4.1.1.3 revslider

Version **6.7.11** is in use. Update to version **6.7.14** or later.

CVSS: 4.4/10 (Medium)

4.1.1.4 filr-protection

Version **1.2.4** is in use. Update to version **1.2.5**.

CVSS: 4.4/10 (Medium)

4.1.1.5 wp-mail-smtp

Version **4.0.1** in use. Update to version **4.1.0** or later.

CVSS: 2.7/10 (Low)

4.1.2 Site Maintenance and Configuration

4.1.2.1 Updates

Ensure WordPress, as well as the third party Plugins and Themes, are updated regularly. Deprecated software no longer receiving updates may need to be removed.

Automatic updates can be enabled, but this can decrease site stability resulting in unexpected issues. For example, an update may introduce compatibility issues with other software implemented on the site.

In case an update introduces stability issues, it is essential to regularly backup the site. These backups will allow you to revert the site back to the last stable version. The integrity of backups should be verified, and ideally backups should be stored in more than one location. If possible, any changes to the website can be tested on a staging site that is not publically accessible before being applied to the live site.

4.1.2.2 Securing Cookies

Verify that the cookies used for authentication use the HTTPOnly flag when possible. With this setting enabled, client side script will not be able to interact with that cookie. This protects the cookie from theft via XSS.

https://owasp.org/www-community/HttpOnly

4.1.2.3 Miscellaneous

I recommend removing DEV's email from URL specified below. There are other ways to discover that he is the Web Developer for this site, but removing the email address from this page would make it more difficult to determine his relationship to the company and find his WordPress account.

https://DOMAIN.com/tab_slider/SUBDIRECTORY/

4.1.3 Using Secure Passwords

Any passwords similar to one that was included in a data breach should be changed.

Use unique passwords for seperate applications to ensure that a compromised password, or some derivitive of it, cannot be leveraged to access additional resources.

The length of a password is the most important factor when it comes to security, and it is generally reccomended to use 14-16 characters. A simple method for creating a strong, yet easy to remember, password is to come up with a unique series of unrelated words, such as “CorrectHorseBatteryStaple”. This is illustrated by the following comic.

https://xkcd.com/936/

4.1.4 WordPress User Accounts

Any user accounts on the website should have the minimum privileges neccesary for their legitimate purpose. For example, an account with Administrator privileges should not be used for day-to-day operations if possible. This would limit the impact of attacks like CSRF. I did not access any accounts, and therefore do not know which privileges they have, but I recommend verifying that accounts are configured in this manner.

https://en.wikipedia.org/wiki/Principle_of_least_privilege

4.1.5 Compartmentalization

Ensure that buisness email addresses are not used in a personal context. This further reduces the likelyhood that a password leaked in a data breach will affect company security.

4.1.6 Third Party Software

The more third party software is used (i.e. Plugins), the larger the site’s Attack Surface will be. Even seemingly innocuous plugins for things like cosmetic changes could at some point contain a vulnerability. It may be beneficial to limit the amount of third party software used where possible, but this is by no means required. Regardless, potential risks can be reduced by employing the aforementioned reccomendations.

4.2 Issues Encountered

This section details ways in which the Penetration Test could have been improved, such as things that I may have missed.

4.2.1 Subdirectory Enumeration

Subdirectory Enumeration is a vital part of a Penetration Test's initial phase. It is used to find files and folders that may not be obvious to users of a site. For example, the subdirectory in the URL **http://example.com/blog/**, is **blog**.

When performing Subdirectory Enumeration on the main domain, **DOMAIN.com**, I got errors from the server rather than accurate results. I was able to somewhat work around this issue by dramatically slowing down my enumeration and spreading it out over the course of several days. Furthermore, I supplemented these results with the website's sitemaps and by crawling the application.

As a result, I believe that I was able to find most, if not all, of the website's endpoints. However, should there be a future engagement, providing me with a list of subdirectories would allow me to verify that all assets are adequately evaluated.

4.2.2 WordPress Plugin Version Detection

I was unable to determine the version for the **tweet-blender** Plugin. There is an XSS vulnerability in versions prior to **4.0.2**, so the version in use should be checked.

The versions of Plugins can be checked by signing in with the Administrator account. If version **4.0.1** or prior is in use, I would recommend uninstalling **tweet-blender**. This Plugin is now deprecated, meaning it is not receiving further updates and the patched version can no longer be downloaded from official sources.

The vulnerability has been given the designation CVE-2013-6342, and an official description can be found at the following link:

https://nvd.nist.gov/vuln/detail/CVE-2013-6342

https://wordpress.org/plugins/tweet-blender/

4.3 Further Actions

Depending on your needs, there are several actions which could be taken on a subsequent Penetration Test for more comprehensive testing. Some, or all, of the following actions may be unneccesary or may not be worth pursuing when weighed against the risk disrupting buisness operations.

- **Account Access**: If a testing account is provided, it can help to determine whether an attacker can leverage low priviledge access to compromise other resources or accounts.

- **Whitebox Testing**: A Whitebox Penetration Test is where the tester is given complete access to Back End resources, such as the source code, a list of all subdirectories, etc. Alternatively, a middleground approach is Greybox Testing, where a pre-determined amount of information is provided. As opposed to the Blackbox Testing that was performed, these types of testing may result in more accurate results and more complete coverage of in-scope assets.

- **Social Engineering & Client Side Attacks**: These attacks require some degree of user interaction, such as clicking a malicious link or opening a malicious PDF. Payloads could be delivered to employees via different means, like through email or the contact form. Although I did not actively perform these attacks, they are consistently a leading cause of compromises for buisnesses of all sizes.

- **Denial of Service Attacks (DoS)**: A DoS Attack would simply prevent legitimate users from accessing your website, but would not compromise its security. There are many variations of DoS Attacks, with some being easier to mitigate than others. Many tests for DoS vulnerabilities will cause disruptions. With this in mind, I do not believe this avenue is worth pursuing unless DoS Attacks become a recurring issue for your website.

5 Appendecies

5.1 Affected Plugins

5.1.1 white-label-cms

https://wordpress.org/plugins/white-label-cms/

CVE-2024-43303:

https://nvd.nist.gov/vuln/detail/CVE-2024-43303

5.1.2 Slider Revolution (revslider)

CVE-2024-37449:

https://wpscan.com/vulnerability/6d7c04cb-b3c1-4003-866b-a3301929e2aa/

5.1.3 filr-protection

https://wordpress.org/plugins/filr-protection/

CVE-2024-43216:

https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/filr-protection/filr-secure-document-library-124-authenticated-editor-stored-cross-site-scripting

5.1.4 contact-form-7-style

https://wordpress.org/plugins/contact-form-7-style/

CVE-2021-24159:
https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/contact-form-7-style/contact-form-7-style-319-cross-site-request-forgery

CVE-2021-4390:

https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/contact-form-7-style/contact-form-7-style-32-cross-site-request-forgery-bypass

5.1.5 wp-mail-smtp

https://wordpress.org/plugins/wp-mail-smtp/

CVE-2024-6694:

https://www.wordfence.com/threat-intel/vulnerabilities/wordpress-plugins/wp-mail-smtp/wp-mail-smtp-401-authenticated-admin-smtp-password-exposure

5.2 Vulnerability Background Information

https://owasp.org/www-community/attacks/xss/

https://owasp.org/www-community/attacks/csrf

https://www.hackerone.com/vulnerability-management/information-disclosure-deep-dive

5.3 Contact Information

You can contact me using the following email address:

Nathan Miller <miller.infosec@gmail.com>